## REMARKS

Claims 1-5, and 7-28 are pending in the present application. Reconsideration of the claims is respectfully requested.

### I. Telephone Interview

Applicants thank Examiner Vo for the courtesies extended to Applicant's representatives during the December 2, 2003 telephone interview. During the interview, Applicant's representatives argued that Anderson does not teach a PCI to PCI bridge as recited in the independent claims. Examiner Vo, however, would not definitively indicate whether the claims define over the Anderson reference. Applicants however, believe the claims in their present state are in condition for allowance. The substance of the interview is summarized in the following remarks.

### II. 35 U.S.C. § 102, Alleged Anticipation of Claims 1-5 and 7-28

The Office Action rejects claims 1-5 and 7-28 under 35 U.S.C. § 102(e) as being allegedly anticipated by Anderson et al (US Patent No. 6,338,119). This rejection is respectfully traversed.

With regard to claims 1, 15 and 22, the Office Action states:

> As for claims 1, 15 and 22, Anderson teaches a method for ensuring that data transferring between a bridge and I/O devices (see figure 1, bridge 108, cache 109, wherein cache 109 is located inside of the bridge and figure 4 step 406 and column 7 lines 45-49, wherein figure 1 discloses I/O devices 118, 120 which are communicating with symmetric multi-processors (SMP) 102 via bridge 108 and vice versa for the SMP to communicate with the I/O devices 117, 120. Further, bridge 108 maintains cache 109 coherency as cited in column 4 lines 63-67 and column 2 line 17 cited wherein the bridge 108 is a PCI to PCI bridge), the method comprising:
>
> monitoring signals from a host bridge for an indication of the state of the data within the cached memory (see figure 1, PCI to PCI bridge 108, cache 109, figure 4 step 406 and 7 lines 6-10, wherein the PCI to PCI bridge determined whether cache 109 is valid or invalid); and
>
> responsive to a determination that data in a portion of the cached memory is stale, clearing at least the portion of the cached memory containing the stale data (see column 4 lines 63-67 and column 5 lines 27-33, wherein one of the embodiment, Anderson teaches the PCI to PCI

bridge maintains cache coherency across L1/L2 caches and cache 109 by clearing buffer memory in cache 109 to make room for new data).
Office Action dated September 16, 2003, page 3.

Claim 1, which is representative of claims 15 and 22 with regard to similarly recited subject matter, reads as follows:

> 1. A method for ensuring that data in a cached memory within a peripheral component interconnect to peripheral component interconnect bridge is fresh, the method comprising:
> monitoring signals from a host bridge for an indication of the state of the data in the cached memory within the peripheral component interconnect to peripheral component interconnect bridge; and
> responsive to a determination that data in a portion of the cached memory is stale, clearing at least the portion of the cached memory containing the stale data. (emphasis added)

As stated in the Response to Office Action dated June 30, 2003, the remarks of which are hereby incorporated by reference, Anderson does not teach monitoring signals from a host bridge for an indication of the state of the data in the cached memory within a peripheral component interconnect (PCI) to peripheral component interconnect (PCI) bridge. In fact, Anderson, in its entirety does not even mention the phrase "peripheral component interconnect to peripheral component interconnect bridge" or "PCI to PCI bridge." This is because Anderson is focused on a PCI Host bridge, not a PCI to PCI bridge.

Anderson does not teach a PCI to PCI bridge, let alone monitoring signals from a host bridge for an indication of the state of the data in the cached memory within the peripheral component interconnect to peripheral component interconnect bridge. However, the Office Action alleges this feature is taught at column 2, line 17 of Anderson. This line merely states "A Peripheral Component Interconnect or PCI Host bridge." Anderson is simply defining the acronym for Peripheral Component Interconnect. In other words, the device may be called a Peripheral Component Interconnect Host bridge or a PCI Host bridge. The allegations made by the Office Action that this particular sentence refers to a PCI to PCI bridge are entirely unsubstantiated, especially considering that nowhere in Anderson is the phrase "PCI to PCI bridge" found and the fact that the figures only show a PCI Host bridge.

Furthermore, the Examiner points to element **109** of Anderson as being a cached memory contained with in a PCI to PCI bridge. Figure 1 of Anderson unequivocally defines element **108** as a PCI Host bridge which contains the cached memory **109**. Figure 1 shows cache **109** and L1/L2 cache **104** and **106**. Thus, cache **109** is not present within a PCI to PCI bridge as recited in claims 1, 15 and 22, but rather cache **109** is located within PCI host bridge **108**. There is no PCI to PCI bridge anywhere in Anderson. Thus, Anderson does not teach each and every feature of claims 1, 15 and 22.

With regard to claims 8, 20, and 27, the Office Action states:

> As for claims 8, 20, and 27, Anderson teaches a method of providing data to an I/O adapter from a bus bridge (see figure 1, bus bridge 108, I/O devices 118, 120 and column 5 lines 8-11, wherein the bus bridge 108 are transferring data to the I/O devices 118, 120 via a conventional adapter), the method comprising:
>
> receiving a request for data from the I/O adapter (see figure 1, bridge 108 and column 7 lines 8-10 and column 7 lines 45-49, wherein figure 1 discloses I/O devices 118, 120 which are communicating with symmetric multi-processors (SMP) 102 via bridge 108 and conventional adapter and vice versa for the SMP to communicate with the I/O devices 118, 120. Further, bridge 108 maintains cache 109 coherency as cited in column 4 lines 63-67);
>
> responsive to a determination that the requested data is contained within a cached memory (see figure 1, cache 109), providing the requested data using the data in the cached memory (see figure 1, PCI to PCI bridge 108, cache 109, figure 4 step 406 and 7 lines 6-10, wherein the PCI to PCI bridge determined whether cache 109 is valid or invalid).

Office Action dated September 16, 2003, pages 4-5.

Claim 8, which is representative of claims 20 and 27 with regard to similarly recited subject matter, reads as follows:

> 8. A method of providing data to an I/O adapter from a peripheral component interconnect to peripheral component interconnect bridge, the method comprising:
>
> receiving a request for data from the I/O adapter;
>
> responsive to a determination that the requested data is contained in a cached memory within the peripheral component interconnect to peripheral component interconnect bridge, providing the requested data using the data in the cached memory.

Anderson may teach a method for providing data to an I/O adapter from a bus bridge as stated in the Office Action. However, claims 8, 20 and 27 do not merely teach

a bus bridge. Rather, claims 8, 20 and 27 clearly recite a peripheral component interconnect to peripheral component interconnect bridge. As set forth above with regard to claims 1, 15 and 22, Anderson does not even mention the phrase "Peripheral Component Interconnect to Peripheral Component Interconnect bridge" or "PCI to PCI bridge". Consequently, Anderson does not teach that the requested data is contained in a cached memory within the peripheral component interconnect to peripheral component interconnect bridge and thus, Anderson does not teach each and every feature of claims 8, 20 and 27.

With regard to claim 10, the Office Action states:

> As for claim 10, Anderson teaches a peripheral component interconnect to peripheral component interconnect bridge (see figure 1, I/O devices 118, 120, conventional adapter and column 5 lines, 1-11, wherein I/O devices 118, 120 are connected to conventional adapter i.e. bridge), comprising:
>
> an interface for sending an receiving data from a PCI to PCI bridge (see figure 1, PCI to PCI bridge 108 and column 5 lines 1-12, wherein data communication are transferring between system memory 110 and I/O devices 118, 120 via PCI to PCI bridge 108 and conventional adapter);
>
> an interface for sending and receiving data from an input/output adapter (see figure 1, PCI to PCI bridge 108 and column 5 lines 1-12, wherein data communication are transferring between system memory 110 and I/O devices 118, 120 via PCI to PCI bridge 108 and conventional adapter);
>
> buffers for storing data (see cache 109, system memory 110);
>
> an interface for receiving signals from the PCI to PCI bridge indicating whether data in the buffers are stale (see figure 4, step 406 and column 7 lines 8-22, wherein the PCI to PCI bridge determines whether cache 109 is invalid i.e. stale); and
>
> logic for clearing stale data from the buffers and retrieving fresh data from the PCI to PCI bridge (see column 4 lines 63-67 and column 5 lines 27-33, wherein one of the embodiment, Anderson teaches the PCI to PCI bridge maintains cache coherency across L1/L2 caches and cache 109 by clearing buffer memory in cache 109 to make room for new data).

Office Action dated September 16, 2003, pages 5-6.

Claim 10 reads as follows:

> 10. A peripheral component interconnect to peripheral component interconnect bridge, comprising:
>
> an interface for sending and receiving data from a PCI host bridge;
>
> an interface for sending and receiving data from an input/output adapter;
>
> buffers for storing data;

an interface for receiving signals from the PCI host bridge indicating whether data in the buffers are stale; and

logic for clearing stale data from the buffers and retrieving fresh data from the PCI host bridge.

Claim 10 recites the feature of a PCI to PCI bridge comprising buffers for storing data. The Examiner argues that Anderson teaches this feature by again pointing to Figure 1. The Examiner states, "cache **109**" and "system memory **110**" in Figure 1 are "buffers for storing data." However, claim 10 does not merely disclose having buffers in general, but instead recites having the buffers for storing data located within the PCI to PCI bridge. In contrast, the alleged "buffers" shown in Figure 1 of Anderson are not located within a PCI to PCI bridge. Cache **109** is present within PCI Host bridge **108**, and system memory is located on system bus **112**. Thus, as evident from Figure 1 of Anderson, cache **109** and system memory **110** are not present within the PCI to PCI bridge. Anderson makes no mention of having a buffer within the PCI to PCI bridge. Consequently, Anderson fails to teach the feature of a PCI to PCI bridge comprising buffers for storing data, as recited in claim 10.

In view of the above, Applicants respectfully submit that Anderson does not teach each and every feature claims 1, 8, 10, 15, 20, 22 and 27 as required under 35 U.S.C. § 102(e). At least by virtue of their dependency on claims 1, 8, 10, 15, 20, 22 and 27, respectively, Anderson does not teach each and every feature of dependent claims 1-5, 7, 9, 11-14, 16-19, 21, 23-26 and 28. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1-5 and 7-28 under 35 U.S.C. § 102(e).

Furthermore, Anderson does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Absent the Examiner pointing out some teaching or incentive to implement Anderson to employ cache memory in a PCI to PCI bridge, one of ordinary skill in the art would not be led to modify Anderson to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify Anderson in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the Applicants' disclosure as a template to make the necessary changes to reach the claimed invention.

In addition to the above, Anderson does not teach or suggest all of the specific features recited in dependent claims 2-3, 11-14, 16-17, and 23-24. For example, with regard to claims 2, 16 and 23, Anderson does not teach or suggest retrieving updated data corresponding to stale data and storing the updated data in a cached memory. The Office Action alleges this feature is taught in the following section of Anderson:

> Within DMA buffer 124, data may be stored in, for instance, 4K page buffers 130 and 132 within 32 lines of data of 128 bytes each. Before L1/L2 cache 102 and 104 can execute a store from processor 102 to a line that is in the shared state in the L1/L2 cache, a separate system bus operation is required in order to inform the other caches to invalidate each cache's copy. Since this is done for each cache line, the processor is slowed down due to the number of repetitive bus operations to clear one page buffer to make room for new data. The present invention sets up a 4Kpage buffer (I/O) so that the buffer may be cleared in one bus operation instead of 32 bus operations.
>
> Typical 4K page buffers are represented by buffers 130 and 132. 4K I/O page buffers, from the present invention, are represented by buffers 134 and 136. Lines of data within the buffers are represented by the blocks within the buffers and a crosshatch within a block represents a shared state. In buffer 130 all the cache lines are shared after the DMA access completes, requiring individual system bus operations for each cache line (32 lines) before the buffer may be cleared. Buffer 132 cache lines are shown as modified allowing data to be written to buffer 132. I/O buffer's 134 first cache line is in a shared state with the remaining lines in a modified state after the DMA access completes as required by the present invention. All cache lines in I/O buffer 136 are in a modified state. In contrast to converting the coherency state of buffer 130 to that of buffer 132, the conversion of the coherency state of buffer 134 to that of buffer 136 requires only the first line in I/O buffer 134 to be changed in order to allow data to be stored to I/O buffer 134. In comparison, converting an I/O page buffer (which only takes one line to change) state would take magnitudes less time than clearing a typical buffer (which requires changing 32 lines to change state). (Column 5, lines 22-55)

This section merely teaches a 4K buffer that is cleared in one bus operation instead of 32 bus operations. Typically, after a direct memory access, all of the lines within the buffer are in a shared state. Thus, each line must be individually changed to a "modified" state in order to write to the buffer. In contrast, Anderson creates a situation in which only the first cached line is in a shared state after a direct memory access. Therefore, only one line must be changed to a "modified" state in order to write to the buffer.

There is nothing in this section or any other section of Anderson that teaches or suggests retrieving updated data corresponding to stale data and storing the updated data in a cached memory. While Anderson may teach retrieving data and storing the data in cached memory, nowhere does Anderson teach that the data being retrieved and stored corresponds to stale data. Thus, Anderson does not teach or suggest all of the features of claims 2, 16 and 23.

With regard to claims 3, 17 and 24, Anderson does not teach or suggest that the signals from a host bridge are sideband signals. The Office Action alleges that this feature is taught in figure 1 and specifically in components 112 and 116. Component 112 is nothing more than a system bus while component 116 is simply a PCI I/O bus. Not only does Anderson not mention the use of sideband signals, it is simply impossible to determine which kind of signals Anderson uses from figure 1. Thus, one can not determine whether Anderson uses sideband signals by simply reviewing the simple depiction of figure 1. Further, Anderson simply does not teach or even suggest the use of sideband signals. Thus, Anderson does not teach or suggest all of the features of claims 3, 17 and 24.

With regard to claim 11-14, Anderson does not teach or even suggest an interface for receiving signals from the PCI host bridge selecting one of a plurality of modes for handling stale data in the PCI to PCI bridge. As set forth above, Anderson does not even mention the phrase "PCI to PCI bridge" and thus does not teach all of the features of claims 11-14. Further, Anderson does not teach a plurality of modes for handling stale data. Applicants agree that Anderson teaches "clearing buffer memory in cache 109 to make room for new data" as recited in the Office Action dated September 16, 2003, page 7. However, Anderson merely teaches one mode for handling invalid data, which is to remove all of the invalid data from the buffer. Thus, Anderson does not teach or suggest selecting one of a plurality of modes for handling stale data in the PCI to PCI bridge as recited in claims 11-14.

Therefore, in addition to being dependent on claims 1, 10, 15 and 22, respectively, dependent claims 2-3, 11-14, 16-17, and 23-24 are also allowable over Anderson by virtue of the specific features recited therein.

## III. Conclusion

It is respectfully urged that the subject application is patentable over the prior art of record and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: December 11, 2003

Stephen J. Walder, Jr.
Reg. No. 41,534
Carstens, Yee & Cahoon, LLP
P.O. Box 802334
Dallas, TX 75380
(972) 367-2001
Attorney for Applicant

SJW/kg